HPTFS: High Performance Tape File System

Xianbo Zhang, David Du
University of Minnesota (DISC)

Jim Hughes, Ravi Kavuri
Sun StorageTek

5/18/2006
Tape Background

• Huge capacity: tape capacity is doubling every two years or 18 months
  – In 2006, one DLT-S4 tape cartridge reaches the capacity of 800 GB native data capacity

• Relatively high streaming rate: tape drive speed is increasing
  – In 2005, Sun StorageTek T10000 drive provides 120MB/s native data transfer rate

• Tape storage has the advantage of low cost per GB, off-site portability and less power consumption compared to other storage solutions
Technologies for Fast Data Location

• Tape cartridge embedded memory chip
  – AIT and LTO tape cartridge embedded memory chip may help obtain data location information without involving tape movement

• Dual mode tape wrap for fast search
  – W/o tape wrapping around drum, tape drive performs fast forward and rewind
  – W/ tape wrapping around drum, tape drive performs write, read and low-speed search
## Tape Capacity/Speed Migration Path

<table>
<thead>
<tr>
<th>Tape technology</th>
<th>2000</th>
<th>2001</th>
<th>2002</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>VXA</td>
<td>33GB, 3MB/s</td>
<td>80GB, 6MB/s</td>
<td>160GB, 12MB/s</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLT Value line</td>
<td>40GB, 3MB/s</td>
<td>80GB, 8MB/s</td>
<td>160GB, 10MB/s</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AIT</td>
<td>100GB, 12MB/s</td>
<td></td>
<td></td>
<td>200GB, 24MB/s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAIT</td>
<td></td>
<td>500GB, 30MB/s</td>
<td></td>
<td>800GB, 45MB/s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Super DLT</td>
<td></td>
<td>160GB, 16MB/s</td>
<td>300GB, 36MB/s</td>
<td>800GB, 60MB/s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTO</td>
<td>100GB, 15MB/s</td>
<td>200GB, 35MB/s</td>
<td>400GB, 80MB/s</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STK9940</td>
<td>60GB, 10MB/s</td>
<td>200GB, 30MB/s</td>
<td></td>
<td>500GB, 120MB/s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBMTS1120</td>
<td></td>
<td>300GB, 50MB/s</td>
<td></td>
<td>500GB, 100MB/s</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Project Motivation

• Tape is needed more than ever due to the explosive data growth rate from content-rich applications and compliance requirements

• To avoid disasters and human errors, critical data are usually backed up to tape and kept off-site
  – Network transmission is not so fast as people expect considering data size of 100’s GB

• Reducing the time to move massive data from disk to tape is critical for the data safety

• Easy to use I/O interface is one of the keys to the further success and broader use of tape storage
System Designed Features

• Providing tape storage access with generic file system interfaces

• Containing user data and corresponding metadata (including directory data) on the same tape.

• Moving data to the final destination – tapes – with streaming speed and does not involve any disk staging
  – Data can be read from tape by application directly without involving disks along the data path.

• Supporting tape drive write sharing with transparent data interleaving
System Architecture

User mode

- op
- glibc

Kernel mode

- VFS
- fuse
- :
- nfs
- ext3

- hptfs /mnt/tape

System call interface

Memory buffer management

- system call interface
- st.o
- scsi_mod.o
- hba

- ...

Glibc

Glibc
Tape Data Residing on Tape

• Tape data is self-contained and light-weighted
  – This is different from any tape file system in the old days

• User data and metadata
  – Each tape maintains three data segments: tape header, user data and metadata
  – Metadata contains object id, start position and end position
  – Metadata can be stored at the end of a tape or in tape cartridge embedded memory chip
Tape Data Layout & Structure

```
struct objid {
    int vol;
    int f_no;
    int b_sp;
    int seq;
};

struct tapemeta {
    char name[1024];
    int f_no;
    int b_sp;
    int b_ep;
    struct objid id;
    struct stat stbuf;
    struct tapemeta *next;
};
```
Write to Tape & Disk Simultaneously
Read while Write

- Disk serves read operations while tape writes in streaming mode
  - Tape read operation is expensive during tape writing process
- Disk can only hold a short period of data while tape library has “infinite” capacity
  - Requires smart purge for high performance
## Example Usage of HPTFS

<table>
<thead>
<tr>
<th>Commands and outputs</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>`[root@oak lib]# ./HPTFS /mnt/tape</td>
<td>Mount tape in write mode at /mnt/tape</td>
</tr>
<tr>
<td>`/home/xzhang/tape w</td>
<td></td>
</tr>
<tr>
<td>`[root@oak lib]# ls -lt *.c</td>
<td>List all C files under current folder (on disk)</td>
</tr>
<tr>
<td>-rw-r-- 1 root root 61725 Jun 2 04:50 fuse.c</td>
<td></td>
</tr>
<tr>
<td>-rw-r-- 1 root root 12461 Jun 2 04:50 helper.c</td>
<td></td>
</tr>
<tr>
<td>-rw-r-- 1 root root 5064 Mar 21 05:37 fuse mt.c</td>
<td></td>
</tr>
<tr>
<td>-rw-r-- 1 root root 3045 Feb 2 2005 mount.c</td>
<td></td>
</tr>
<tr>
<td>`[root@oak lib]# cp *.c /mnt/tape</td>
<td>Copy all C files from disk to tape</td>
</tr>
<tr>
<td>`[root@oak lib]# fusermount -u /mnt/tape</td>
<td>Write out metadata to tape and umount tape</td>
</tr>
<tr>
<td>`[root@oak lib]# ./HPTFS /mnt/tape</td>
<td>Mount tape in read mode at /mnt/tape</td>
</tr>
<tr>
<td>`/home/xzhang/tape r</td>
<td></td>
</tr>
<tr>
<td>`[root@oak lib]# ls -lt /mnt/tape</td>
<td>List all C files on tape media</td>
</tr>
<tr>
<td>-rw-r-- 1 root root 61725 Aug 15 23:55 fuse.c</td>
<td></td>
</tr>
<tr>
<td>-rw-r-- 1 root root 5064 Aug 15 23:55 fuse mt.c</td>
<td></td>
</tr>
<tr>
<td>-rw-r-- 1 root root 12461 Aug 15 23:55 helper.c</td>
<td></td>
</tr>
<tr>
<td>-rw-r-- 1 root root 3045 Aug 15 23:55 mount.c</td>
<td></td>
</tr>
</tbody>
</table>
Performance Evaluation

Setting A: slow host with PIII 500Mhz cpu, 256 MB + STK 9840A tape drive
Setting B: faster host with four Intel(R) XEON 2.40GHz cpu’s, 3GB + STK 9940A tape drive

Main observations:
• User applications directly write/read data to/from tape without the knowledge of tape storage
• Support concurrent writes nicely
• Stream tape drive if enough data are provided
Part of the Performance Results

Table 2. Tape write performance (MB/s, tape block size=256KB)

<table>
<thead>
<tr>
<th>Degree of concurrency</th>
<th>Setting A rate</th>
<th>Setting B rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Stdv</td>
</tr>
<tr>
<td>2</td>
<td>24.148</td>
<td>0.433</td>
</tr>
<tr>
<td>3</td>
<td>24.222</td>
<td>0.392</td>
</tr>
<tr>
<td>4</td>
<td>24.169</td>
<td>0.373</td>
</tr>
</tbody>
</table>

Note: write speeds of Setting A and B are rated as 29.759 MB/s and 37.604MB/s respectively
Tape Random Read Performance from PostMark

Table 7. Tape read performance with PostMark (MB/s, tape block size=256KB)

<table>
<thead>
<tr>
<th>Degree of interleaving</th>
<th>Setting A rate</th>
<th>Setting B rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Stdv</td>
</tr>
<tr>
<td>1</td>
<td>1.750</td>
<td>0.021</td>
</tr>
<tr>
<td>2</td>
<td>1.835</td>
<td>0.049</td>
</tr>
<tr>
<td>3</td>
<td>1.695</td>
<td>0.034</td>
</tr>
<tr>
<td>4</td>
<td>1.470</td>
<td>0.127</td>
</tr>
</tbody>
</table>
Tape Random Read Performance with PostMark (1,000 files and 100 read operations)
## File Signature Comparison

### Table 3.9: “Screenshot” and annotation

<table>
<thead>
<tr>
<th>Commands and outputs</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>oak%./HPTFS /mnt/tape w</code></td>
<td>Mount tape in write mode at /mnt/tape</td>
</tr>
<tr>
<td><code>oak% ls -lt *.c</code></td>
<td>List all C files under current folder (on disk)</td>
</tr>
<tr>
<td><code>-rw-r-- 1 root root 61725 Jun 2 04:50 fuse.c</code></td>
<td></td>
</tr>
<tr>
<td><code>-rw-r-- 1 root root 12461 Jun 2 04:50 helper.c</code></td>
<td></td>
</tr>
<tr>
<td><code>-rw-r-- 1 root root 5064 Mar 21 05:37 fuse_mt.c</code></td>
<td></td>
</tr>
<tr>
<td><code>-rw-r-- 1 root root 3045 Feb 2 2005 mount.c</code></td>
<td></td>
</tr>
<tr>
<td><code>oak% cp *.c /mnt/tape</code></td>
<td>Copy all C files from disk to tape</td>
</tr>
<tr>
<td><code>oak%fusermount -u /mnt/tape</code></td>
<td>Write out metadata to tape and umount tape</td>
</tr>
<tr>
<td><code>oak%./HPTFS /mnt/tape r</code></td>
<td>Mount tape in read mode at /mnt/tape</td>
</tr>
<tr>
<td><code>oak% ls -lt /mnt/tape</code></td>
<td>List all C files on tape media</td>
</tr>
<tr>
<td><code>-rw-r-- 1 root root 61725 Aug 15 23:55 fuse.c</code></td>
<td></td>
</tr>
<tr>
<td><code>-rw-r-- 1 root root 5064 Aug 15 23:55 fuse_mt.c</code></td>
<td></td>
</tr>
<tr>
<td><code>-rw-r-- 1 root root 12461 Aug 15 23:55 helper.c</code></td>
<td></td>
</tr>
<tr>
<td><code>-rw-r-- 1 root root 3045 Aug 15 23:55 mount.c</code></td>
<td></td>
</tr>
<tr>
<td><code>oak% cp /mnt/tape/fuse.c ./fuse_1.c</code></td>
<td>Copy <code>fuse.c</code> from tape to disk as <code>fuse_1.c</code></td>
</tr>
<tr>
<td><code>oak% openssl</code> OpenSSL&gt; sha1 fuse.c SHA1(fuse.c)= c8ab9be7c2edc1128db66f877b40ceeafffb74f6<code> OpenSSL&gt; sha1 fuse_1.c SHA1(fuse_1.c)= c8ab9be7c2edc1128db66f877b40ceeafffb74f6</code></td>
<td>Comparing the original <code>fuse.c</code> on disk to <code>fuse_1.c</code> copied from tape with SHA1.</td>
</tr>
</tbody>
</table>
“Infinite” Online Backup/Archive Storage

HTTP Server
CGI
/export/tape1
/export/tape2
...
/export/tapen

MySQL server
Tape metadata and media information database

Internet

Client side encryption

client 1
client 2
...
client n

Infinite" Online Backup/Archive Storage

HTTP Server
CGI
CGI
/exports/tape1
/exports/tape2
...
/exports/tapen

DMDS

MySQL server

HPTFS
/tape1
/tape2

DMDS

HPTFS
/tape1
/tape2

DMDS

data path
control path
host
Conclusions

• HPTFS provides generic file system interface for tape data access: writing to tape is as easy as writing to disk
• Provides tape drive sharing with high performance
• Built over HPTFS, software for backup and HSM can be made simpler
• Potential to embed HPTFS functionality into tape drive totally changing tape access paradigm
• OSD interface can be easily provided over HPTFS